The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 10||Pages|| 24-34||2013||
ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805

Detection of Botnet Multi-Stage Attack By Using Alert Correlation Model

[1,]Mohammed Alnas , [2,]Abdalla M. Hanashi , [3,]Elmabruk M Laias
[1,]*Computer Department, Faculty of Science Alzituna Universit, Tarhona, Libya*
[2,]*Computer Department, Faculty of Engineering Azzawia Universit, Azzawia, Libya*
[3,]*Computer Department, Faculty of Science Omar Al-Mukhtar UniversityDerna, Libya*

----------**ABSTRACT**----------
Network Intrusion Detection Systems (NIDS) are considered as one of the essential mechanisms to ensure reliable security. Intrusive model is used in signature-based NIDS by defining attack patterns and applying signature-matching on incoming packets. However, detection of novel and multi-stage attacks are not efficiently achieved by the signature-based systems. This is due to lack of mechanism to perform sophisticated analysis to identify relationship between attack events. Hence, the systematic analysis of attack initiation has become a stressing demand in current research. Alerts correlation techniques have been widely used to provide intelligent and stateful detection methodologies. This is to understand attack steps and predict the expected sequence of events. However, most of the proposed systems are based on rule –based mechanisms which are tedious and error prone. Other methods are based on statistical modeling; these are unable to identify causal relationships between the events.In this paper, we have identified the limitations of the current techniques and propose a model for alert correlation that overcomes the shortcomings. An improved "require/provide" model is presented which established a cooperation between statistical and knowledge-based model, to achieve higher detection rate with the minimal false positives. A knowledge-based model with vulnerability and extensional conditions provide manageable and meaningful attack graphs. The proposed model has been implemented in real-time and has successfully generated security events on establishing a correlation between attack signatures. The system has been evaluated to detect one of the most serious multi-stage attacks in cyber crime - Botnet. Zeus Botnet is analyzed within the realm of simulated malicious activities normally used by cyber criminals. The system has efficiently established a correlation in attack behaviors and has generated an attack map. The map can be used to discretely analyze the correlated attack activities which in other case may go undetected thus facilitating the multi-stage attack recognition process.

KEYWORDS: *Network intrusion detection systems, Alerts correlation, multi-stage attack, Alert Correlation, Botnet*

Date of Submission: 26 September 2013 Date of Publication: 20 October 2013

I. INTRODUCTION

Malicious attacks by intruders and hackers exploit flaws and weaknesses in the deployed systems. This is done by several sophisticated techniques which cannot be prevented by traditional security measures. Fame is now no more the hacker's destiny; their efforts are have profitable gains from malicious activities. The current trends in cyber attacks are hidden, coordinated and slow-and-low. NIDS are considered to be important security tools to defend against such threats. The effectiveness of any NIDS depends on its ability to recognize different variations of cyber attacks. The current implementation of intrusion detection systems (commercial and open-source) is employing signature-based detection mechanisms. In addition to these, few statistical techniques are also used for detection process. The main task of signature-based systems is to inspect the network traffic and perform pattern matching to detect attacks and generate alerts. The systems generates large number of alerts everyday and make the job of administrator difficult as the person has to sift the entire alert log to find out actual attacks. Quality of these alerts is also debatable particularly if the majority is false positives. For this reason, high-level and real-time analysis techniques are needed. Potentially a more suitable way of analysis is discovering logical connections between isolated alerts. It has been practically identified that most of attacker activities consists of multiple steps (attack scenario) and occur in a certain time (attack window). Identification of such strategy can lead to the recognition of attack intensions and also prediction of unknown attacks. Some simple analysis tools have been developed to generalize these alerts based on attack classes [3].

In recent years, Botnets [4][24][29] have been one of the most serious multi-stage attacks against web technologies to obtain access to computer systems and to control them remotely. Botnets are collections of software agents installed in compromised machines known as zombies, and commanded and updated by a bot master using Command & Control channels (C&C). Groups of cyber organized criminals have employed these techniques widely to achieve distributed attacking platforms in order to launch various planned attacks against online systems. They are used for Distributed Denial of Service (DDoS), email spam, phishing attacks, data theft, and malware infections. Botnets use various attack vectors such as bogus scripted emails and attractive malicious websites. They can also exploit protocol vulnerabilities and make use of buffer overflows. Malware is installed in the vulnerable machine remotely, while Bot master uses C&C in an organized way to achieve personal gains.The attacker searches for a vulnerable system connected to the Internet to exploit and to obtain the maximum privileges exploiting different vulnerabilities. Social engineering is another vector of infection using emails, malicious websites and instant messages. A malicious code is installed in the victim machine which in turn connects to C&C server to get updated and controlled by the Bot master. Hence, a new member joins the team and now it is ready to involve in attacking new victims using facilities provided by the Bot master. Capabilities gained by attacker involve the target machine resources, bandwidth, and processing power which can be used for financial gains.The rest of this paper is organized as follows: section 2 presents the related work and section 3 explains the problem scope. Section 4 provide a background of provides/requires model. Section 5 gives an overview of MARS model [8] and in Section 6 we analyze Botnet attack. In section 7 we present the experimental results, and then we conclude in section 8.

II. RELATED WORK

Alerts clustering and correlation techniques have been employed to provide a global view of attacker's behavior by analyzing low-level alerts produced by the IDS sensors. The main objective of alerts correlation is to build an abstract modeling of alerts by generalizing the detected events instead of the current specific modeling. The constructed inference will progress even in case of unforeseen attacks. Different approaches have been utilized to build the correlation models[5], and can be categorized into three main disciplines: probabilistic approaches, scenario-based approaches and pre/post conditions approaches. The probabilistic approaches are inspired from anomaly-based intrusion detection systems where prior knowledge is not required. In this category, relations between incurred events are computed statistically providing automatic knowledge acquisition. Data mining, clustering, association rules techniques are examples of these approaches. The work in [2] presented a probabilistic approach to provide unified mathematical framework that perform a partial matching of features. Features are extracted and minimum similarities are computed and weighted. K Julisch in [14] proposed alarm clustering to discover the root causes of different alarms. The aim was to reduce the volume of alarms to manageable size. Even though, these methods are useful for alert fusion and statistical purposes but they fail to discover the causal connection between alerts.Recently the efforts in [1], [13] and [30] employed different data mining algorithms for real-time correlation to discover multi-stage attacks. Off-line attack graph is constructed using manual or automatic knowledge acquisition and then attack scenarios are recognized by correlating the collected alerts in real-time.

The incoming step of an attack can be predicted after detection of few steps of attack in progress. In [30] association rule mining algorithm is used to generate the attack graph from different attack classes based on historical data. "candidate attack sequences" are determined using a sliding window. In [16] AprioriAll algorithm which is a sequential pattern matching technique is used to generate correlation rules based on temporal and content constraints. The [16] adopted a classical sequential mining method GSP [20] to find the maximal alerts sequence and then to discover the attack strategy. The limitation of their work is the use of only attack class and temporal as features.On the other hand, scenario-based modeling is based on manual knowledge acquisition that specifies intrusion steps by experts. Scenario libraries are used to build the model and to discover the logical connections between alerts. LAMBDA [10] is an intrusion specification language to describe the conditions and effects of an intrusion in connection to the variable state of the victim system. Similarly, in STATL [21] language, sequence of events conducted by the attacker can described to express multi-stage attack. However, these approaches need a manual description of possible attacker's behavior and if a single step is missed the whole behavior go undetected. The third category is the pre/post conditions techniques which are based on the notion that the older alerts prepare for the later ones. These approaches require specifying the criterion used to discover the relations between alerts and the weights of such relations. Early, [22] proposed a "require/ provide" capabilities model using attack specification language "JIGSAW". However, the exact matching between "require" and "provide" conditions is employed causing different variation of the same behavior is not detected.

[9] proposed MIRADOR correlation approach for alert clustering, merging and then correlation. Explicit correlation of events based on security experts is used to express the logical or topologic links between events. Attack is specified using five fields and based on the language of LAMBDA [10]. Partial matching techniques are adopted to build the model. In addition to explicit correlation, implicit correlation is used to overcome possibly missing events.Authors in [18] and [19] proposed alert correlation model based on prerequisites and consequences of individual detected alerts. A knowledge database "Hyper-alert Type Dictionary" contains rules that describe the conditions where prior behaviors prepare for later ones. Attack strategy is represented as a Directed Attack Graph (DAG) with constraints on the attack attributes considering the temporal order of the occurring alerts. The nodes of the DAG represent attacks and the edges represent causal and temporal relations. Similarities between these strategies are measured to reduce the redundancy. A technique of hypothesizing and reasoning about missing attacks by IDS is presented to predict attribute values of such attacks. The significance of their work is the reduction of the huge number of security incidents and to report a high-level view for the administrator. However, the proposed system is useful as a forensic tool where it perform offline analysis. In addition, building the knowledge database containing rules of the applied conditions is a burdensome. However, authors have not provided a mechanism to build the Hyper Alert dictionary. Also, the generated graph is huge even with medium size datasets.

In other respect [26] and [27] proposes a combination of statistical and knowledge-based correlation techniques. Three algorithms are integrated based on assumption that some attack stages have statistical and temporal relations even though direct reasoning link is not existent. Bayesian-based correlation engine is used to identify the direct relations among alerts based on prior knowledge. In contrast to previous approaches, knowledge of attack steps incorporates as a constraint to probabilistic inference to avoid the exact matching of pre and post conditions. Causal Discovery Theory-based engine is developed to discover the statistical of one-way dependence among alerts. In addition, Granger-Causality-based algorithm is used by applying statistical and temporal correlation, to identify mutual dependency. However, the problem of selection time window for temporal correlation is still an open problem. Attackers can exploit the slow-and-low attack to avoid detection. Attack prediction also relies on prior knowledge where zero-day attack is not detected.Although the past techniques dealt with reducing the massive number of collected data by NIDS, however there are many limitations. First, the analysis of attack strategy recognition is too complex especially if the task broadens to predict the unknown steps. Knowledge-based approaches are more accurate due to rules matching mechanism which are built based on experts' knowledge, but it needs more efforts to provide precise rules. Statistical and temporal analysis techniques are unable to detect causal relations among events, but they don't require prior defined rules. Adoption of such systems in real-time is still an open question, where most proposed systems have been tested in offline fashion or in a low volume traffic environment. The huge number of detected events leads to graph explosion as in [18][19]. Moreover, missing attacks by the IDS can result in separate scenarios related to the same attack. Attackers also exploit the attack sliding window used in most approaches by performing slow-and-low attack.

Alerts correlation modeling has to provide a type of intelligence for attack strategy recognition. A framework consists of several components is needed to enjoy capabilities of different approaches. A combination of knowledge-based, statistical and temporal based, data mining and machine learning can incorporate to provide more intelligent system. In this paper we propose a novel approach to overcome the limitations of the past techniques. Attack strategy recognition cannot be implemented in a single stage or using a single component.In this paper, our work aims to build an improved correlation model based on "requires/provides" conditions techniques [18], [19], [10], [11], [9] and [21]. We have selected this approach for several reasons. First, instead of specification of the whole steps of the attack scenario, only the specification of "requires" and "provides" conditions of an event is required. This provides a flexible approach particularly if a partial satisfaction of correlation is employed. Second, even if the attacker starts the attack from advanced steps, the behavior is still detected. Similarly, if an alert is missed and the scenario is divided into different graphs, these sub-scenarios can be correlated. Third, the model is expandable to incorporate other mechanisms such as probabilistic approaches. MARS [8] has initially proposed to combine two engines: online and offline, and two mechanisms: high quality knowledge-based and statistical-based correlation. In addition, the proposed model employs various tools that help the administrator to recognize multi-stage attacks and attackers behaviors. An overview of the system will be presented in section 4.

III. PROBLEM SCOPE

It has been identified from cyber-security field that well-planned attack consists of number of stages conducted in a temporal order. True alerts belong to intrusion generated by the IDS systems are not isolated; they also reflect the sequential pattern of the attacker. However, IDS systems consider these alerts as individual

events and report that to the administrator with huge amount of alerts most of them are false positives or not critical for the protected system. A high level view of these incidents can assist to recognize attacker's plan and take a rapid action to maintain the security state. Moreover, IDS systems due to their limitations cannot detect all variation of unseen attacks. However, the alert correlation systems can predict the upcoming attack based on the pervious behaviors of attackers. Also, False alarms can be excluded because they are often isolated and non-critical events.

In order to achieve this task it is required that the correlation approach considers:

- Real-time or at least near real-time correlation that inspect the incoming alerts and correlate them to the older ones. However, it is a challenging task particularly if we consider the scalability, the huge amount of alerts and the speed of the current implementation of communication networks. Authors in [18] and [19] developed TIAA system that perform the correlation in memory using nested–loop mechanism and [30] proposed queue graph mechanism. However, they have not provided any evaluation in high-speed networks to assess the system scalability.
- Recognition of missed attack by the IDS which will cause a division of scenario or graphs into separate ones. The correlation system has to be able to correlate isolated scenario using implicit correlation. This mechanism also, can be used to predict unknown attacks by hypnotizing the expected step which can be variations of known attacks.
- Slow-and-low attacks conducted by skillful attackers to avoid detection. Most of the implemented system uses a sliding window to avoid graph explosion and hence very old events are ignored. However, determination of the value of sliding window is also critical to provide higher detection rate. Ignoring old events can result in the success of a dangerous intrusion attempt.
- Alert verification where not all alerts are critical and they have different effects on the system. This mechanism will reduce the huge number of correlated alerts by focusing on the significant ones.
- The configuration of the protected system can incorporate to reduce false positives and provide higher meaningful and accurate results. Host response can also be involved to shift the focus to the critical events.

  The main contribution of this work is a part of the development and evaluation of the proposed framework for alert correlation system that meets these requirements.

IV. REQUIRES/PROVIDES MODEL

It has been proposed by [22] in inspiration from network management systems to deal with network faults. Cyber attack is described in two components: capabilities and concepts. The idea behind this model is that multi-stage intrusion consisting of a sequence of steps performed by an attacker; the later steps are prepared by the early ones. Target system information collected from scanning or port mapping, are advantages acquired to choose which exploit can be used. Capabilities are defined as general description of the conditions required or provided by each stage of intrusion. In other words, the system state that must be satisfied in order to launch an attack. For instance, a successful Trojan injection requires some particular services running in the target systems and an existence of vulnerabilities Formally, capabilities are a higher level of intrusion abstraction that specifies the system state after each attack attempt. Concepts are abstracts of system states that involved in multi-stage attack scenarios. Attacker uses the capabilities gained by some of his early actions to generate some new capabilities. System state incorporates in attack scenarios if instances of concepts have "required" and "provided" conditions matched.

The capability model proposed by [25] is also based on "requires/provides" model for logical alert correlation. The authors used different properties of capabilities. An attack model is presented to build blocks of capabilities in a multilayer fashion with more expressive definition. [9], [10] and [11] have used "requires/provides" model using the concept of predicates which are similar to capabilities.

Our model is a variation of the "requires/provides" model but it is different in the following aspects:

- Different definitions for capabilities and concept are employed to overcome the limitations expressed in other approaches; these will be discussed in section 5. The work in [22] used very detailed specifications language called JIGSAW to describe attack scenarios. A complete satisfaction of "required" and "provided" conditions is necessary to correlate two alerts and that will fail in case of broken scenarios. However; [18] and [19] have adopted a partial satisfaction technique which is also implemented in our model. The main concern with their approach is the high rate of false positives and possibly a huge graph will be created. We have managed to overcome this limitation by using three techniques: well-defined capabilities, accumulated aggregation and alert maintenance.
- Real-time processing approach for correlation, aggregation and event generation. The security officer can monitor the attack progress which is displayed as an intrusion graph. An event is triggered once at minimum

two alerts are correlated and any additional related alert based on its attributes will join the same event.

- Some parameters are not considered in other approaches are proposed such as vulnerability abstraction, attack direction, and administrator experience.
- Online and offline graph reduction algorithms after correlation and aggregation to provide a manageable graph.

V. MARS MODEL REVIEW

This section presents briefly the knowledge base of MARS model that generates rules to correlate high-level alerts called Meta-Alerts. As stated earlier, our model is derived from "provides/requires" model using different definitions of the model components. The proposed model for the knowledge base consists of three sets:

- Capability C: This specifies a higher level of abstraction of intrusion model. Intrusion attempts are expressed in terms of a set of "required", "provided", and extensional "provided" conditions of a given alert.
- Meta Alert (M-Alert) concept MC: This specifies the related capabilities of a given Meta-Alert. "Required" and "provided" conditions for each M-Alert are coded in language of capabilities.
- Meta-Alert M: a higher level of abstraction of an alert. This can be generated from various IDS sensors. In our case, we use Snort[23] as the main IDS, so Meta-Alert will be elementary alert received from Snort . However, different M-Alerts will be aggregated in different occasions during the correlation process.

Definition1. A M-Alert concept MC is an abstraction of elementary alerts generated by IDS defined by a set of (Arguments, Required Conditions, Provided Conditions, Extensional Provided Conditions, Vulnerability, Intrusion direction, and Experience) where:
Arguments $[r_1, r_2, \ldots r_i] \rightarrow r$: are a set of associated attributes such as source and destination IP addresses.

Required Conditions R : are a set of pre-conditions specified in a form of capabilities with variable of Arguments.

Provided Conditions P : is a set of post-conditions specified in the form of capabilities with variable of Arguments.

Extensional Provided Conditions EP: are a set extended Provided Conditions as a result of implicit relations between capabilities in a form of capabilities with variable of Arguments.

Vulnerability V: is a description of state of the target host or network with variable of Arguments.

Intrusion Direction D: is a description of attack direction (0: source address, 1: destination address, 2: bidirectional)

Experience EX: is description of the security officer's feedbacks in different situations.

The provided P conditions are extended to involve possible extensional provided conditions EP to broaden the correlation mechanism. This is the result of possible implicit correlation between alerts based on interdependencies between capabilities. This mechanism will be useful in two cases, the first: suppose the attacker ignore some steps because he has already obtained some knowledge about the target system. Then, there is no need for creating unnecessary noise that may lead him to be noticed. The result will be a broken scenario and most of proposed correlation system fails to correlate this sort of sequences. The second case: it has been identified that NIDS systems miss some attacks because of absence of its signatures or if the system experience high speed traffic that the NIDS is unable to process all packets.The information provided by elementary alerts does not reflect the actual state of the target system. For this reason, we proposed additional information about the vulnerability parameters and the state of the victim to produce more realistic correlation. The vulnerability knowledge, which can be acquired using tools such Nessus[17], will ignore insignificant alerts from correlation process to reduce the complexity of the resulting graph. A produced huge graph with false positive correlation in [18][19] is avoided. In addition, attack direction and administrator experience are adopted to raise the accuracy and hence, to lower the false alarms.

Definition. 2 An M-Alert instance m is defined as a set of instances of M-Alert concept MC by substituting the associated values in Arguments tuple considering the time constraints (start-time and end-time).

Definition. 3 Given a M-Alert concept MC and an M-Alert instance m, the R(MC), P(MC), EP(MC), V(MC), and EX(mc) sets are the sets of all Capabilities C. Given an M-Alert instance *m*, the R(m), P(m), EP(m), V(m), and EX(m) sets are the capabilities by mapping the values to the corresponding Arguments in MC considering the time constraints.

Definition 4. Given a pair of M-Alert instances m : m_1, m_2 ordered temporally in the following time slots respectively:

m_1 *:* t_{s1} and t_{e1}

m_2 *:* t_{s2} and t_{e2}

where t_s is the start time, and t_e is the end time.

m_1 is correlated with for m_2 if:

1- There exists at least one common Capability C in R(m_2), P (m_1) and EP (m_1).

2- Satisfaction of V(m_2), EX(m_2), and D(m_2) constraints.

3- P(m_1).t_{e1} >= R(m_2).t_{s2} AND

EP (m_1).t_{e1} >= R(m_2).t_{s2}

The partial matching mechanism has been used to avoid the hard-coded correlation as in scenario-based methods and the explicit relationship employed in other "requires/provides" approaches.

Definition 5. Correlated Attack Graph CAG(N,E) is defined as a Directed Acyclic Graph (DAG) consisting of a set of nodes N connected by edges G. Nodes n_1,n_2,n_3, ….,n_i □ □N represents the M-Alert set and edges g_1,g_2,g_3,….g_j represent the "provide" relationship. Formally, Let M alerts represent some exploits E discovered in a system, and C to be set of capabilities represents the relevant security conditions. To express the relationship between system conditions and possible exploit there are two relations:

R □□C X E

P □ E X C , EP □ E X C

And the correlated attack graph is

CAG(M□ □ □C, R □ □ □P□ □EP)

For this reason, we can say that the relationship between system conditions, exploits, and alert instances is a logic correlation expressed in AND and OR. OR logic is used between the conditions required and provided and AND is used to satisfy particular instances of the target system. The alert correlation algorithm is shown in Figure 1.

Algorithm: Alert Correlation

Input: elementary alerts generated by the IDS

Output: Correlated Attack Graph $CAG(\mathcal{N},\mathcal{G})$

Methods:

1- **Let** $CAG(\mathcal{N},\mathcal{G})$ = null

2- **Map** elementary alerts to M-Alerts instances ($m_0, m_i, \ldots\ldots, m_i$)

3- **Let** m_0 an instance of isolated $\mathcal{M}$

4- **For** k=1 to i

If a. at least one $\mathcal{R}(m_{i+1}) \in \mathcal{P}(m_i)$

$\cup\ \mathcal{R}(m_{i+1}) \in \mathcal{EP}(m_i)$

b. $\mathcal{V}(m_{i+1})$, $\mathcal{V}(m)$, $\mathcal{EX}(m)$, and $\mathcal{D}(m)$ are satisfied.

c. $\mathcal{P}(m_i).End\_time >= \mathcal{R}(m_{i+1}).Start\_time$

$\cup\ \mathcal{EP}(m_i).End\_time >= \mathcal{R}(m_{i+1}).Start\_time$

Then

Add $CAG(n_{m_i}, n_{m_{i+1}})$

5- Return $CAG(\mathcal{N},\mathcal{G})$

FIGURE 1. Correlation algorithm

MARS tools have been designed and implemented using C++ language and MSSQL database. Figure 2 shows the implemented system architecture. System details are not described here due to space constraints. We have evaluated MARS using DARPA2000 in [15] and it has achieved improved results.

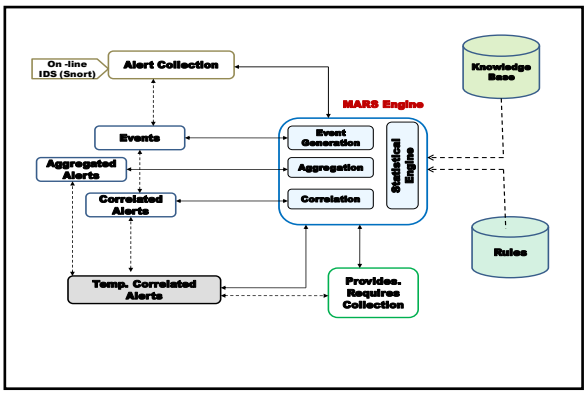

FIGURE 2. MARS system Architecture

VI. CASE STUDY: BOTNET

Botnet attack is a multi-stage and coordinated process; and to detect such activity we need to obtain the whole picture of the attacker behavior. IDS systems network-based and host-based are able to detect some attacks based on their signatures or protocol analysis. However, detected events are treated as isolated activities and uncountable variations of Botnets are discovered every day. Attackers tend to change their fingerprints to avoid detection by IDS rules despite the general behaviors are similar. Even though, the IDS system misses some attack involved in Botnet activity, network administrator is still aware of the global view of a suspected Botnet behavior. In addition, according to several behavior analysis [4][24], Botnets communications and activities are similar regardless of the common name of any used malicious software. For instance, Zeus, Kneber, and bredolab [4] are variations of the same malicious modular Botnets. Even though, different Botnets have been identified in security analysis field, almost all follow similar steps which are known as Botnet lifecycle. These sequences are shown in Figure. 3 and summarized as follow:

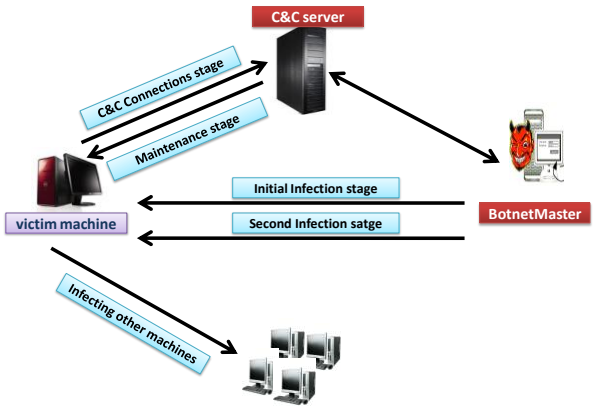

FIGURE 3. Botnet lifecycle

1. Initial infection stage: This stage involves scanning for systems running vulnerable services or responding to backdoors.
2. Second infection stage: Remote malicious code is loaded and software is installed in the target machine using one or more available attack vectors. The infected system is ordered to download the actual Botnet software from a dedicated Bot server. Then, the code is executed and the machine becomes a botnet member.
3. Connection to C&C stage: The infected machine connects to the attacker and receives commands to be configured and updated using C&C channels over IRC or HTTP. In this stage, the actual Botnet activities started.

4. Attacking other machines stage: scanning activities are maintained to discover un-patched and vulnerable systems to launch further possible infections.
5. Maintenance stage: upon the capabilities of the target machine the attacker commands the Botnet members to download binaries, to connect to another C&C server and to involve in attacking some other victims. The attacker also has to be certain that all members can be reached using Fast Flux DNS technique [6] to hide malicious code deliveries under all dynamic network conditions.

Zeus [4] Botnet is one of the emerging modular Botnets reflecting the darkness of cyber crime world, first identified in 2007 [4]. It is also known as banking crime ware and motivated initially to steal banking credentials and account information. Zeus has some abilities includes stealing data submitted by HTTP forms, emails and FTP account information, stealthy injection of HTML on the fly, and all redirection activities to trap victims. It is a package of software with GUI and its builder is responsible to create all necessary files such as executable, PHP files and SQL templates with a straight forward manner. We have installed an older version of Zeus as the new versions are sold by licence, on one of our machines in our lab in an isolated network. We have followed the typical scenario in real life simulating the traffic communications between the Bot master and the victim machines. The simulated network is monitored by Snort and MARS engine. Snort is configured with all rules enabled including: VRT, bleeding-Edge, Community, and Emerging Threat rules (ET).

VII. EXPERMINENT AND RESULTS

In this paper, simulating Zeus Botnet attack has used to test the detection accuracy of the proposed model. We have pursued Botnet scenario as occurs in real network as described later in this section. Network traffic has been recorded in a pcap file for further analysis. Then we have injected the produced pcap file with 200MB of other traffic consisting of: normal traffic, background traffic, and some malicious traffic. We have also, modified some fields of the injected noise traffic to be synchronized. The attack steps are as follow:

A. The attacker starts to perform scanning looking for vulnerable systems in order to exploit or to install a backdoor in the target machine. In this scenario, the attacker will use a new identified application flaw which is CVE-2010-0188 [7], Adobe Reader in versions earlier than 9.3.1. An embedded executable code Launch command can be used to infect the target machine. Metasploit[12] is used to perform this job by copying a malformed malicious PDF documents to the victim machine. Snort has triggered two signatures related to scanning activity and three other signatures in connection to Shellcode and CVE-2010-0188 vulnerability. As shown in Figure 4 the five alarms are correlated in a sequence. This scenario is not necessary to be Botnet activity because it can be any other attempt to obtain system access.

sid: 1394 SHELLCODE x86 inc ecx NOOP
sid: 16490 SPECIFIC-THREATS Adobe Reader malformed TIFF remote code execution attempt
sid: 15013 WEB-MISC Adobe Portable Document Format file download attempt

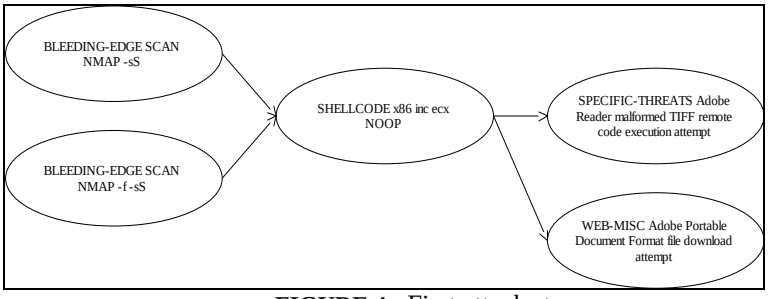

FIGURE 4. First attack stage

B. The target host is infected and starts to connect to C&C server to download binaries and configuration files. An HTTP GET request is sent to C&C server to obtain an encrypted configuration files. While these files are encrypted and their names and the URL are random, it is very difficult for Snort and all other signature-based IDS to detect such files. However, an alarm has been triggered in this stage recognizing the name of the configuration file. These signatures have been added to Snort VRT in version 2.8.6.1 in July 2010 [23]

sid: 2008100 ET TROJAN PRG/ Zeus InfoStealer Trojan Config Download
sid:16912 BLACKLIST URI request for known malicious URI - net/cfg2.bin

The previous signatures are one of a group of signatures to block some suspicious URI request containing malicious websites tracked by Zeus Tracker [28].

C. Followed by the configuration files, an HTTP POST request, sent to the same C&C server in the second stage to fetch PHP files and again the data in POST request is encrypted. Snort fired an alarm similar to the alarms in the second stage but with different URI.

sid:16929 BLACKLIST URI request for known malicious URI - gate.php?guid=

D. Despite the previous two steps can be performed without Snort response using some obfuscation techniques, this stage can be identified. The server response for the last step contain some recognized behaviour, that's the string "Content-Type:text/html" and the actual data is not HTML or other legitimate formats. Actually, there is a signature in Snort that can catch this piece of traffic, which is sid:16460, but it is deleted due to false positives concerns as this case may exist in normal traffic. So, if we have a system that recognises false positives generated by Snort, and this is the case for MARS system, this alert will be ignored if they are not involved in real attack scenario. For this reason, we have enabled the 16460 rule to provide more information and in case of isolated false alarm, it is will not contribute in the attack picture. In addition, Snort has triggered some other alerts based on ET rules that identified some small binaries downloads and these are some suspicious behaviours have to be noticed. The correlated and aggregated alerts' sequence involved in this stage and the previous two stages are shown in Figure 5.

sid:16460 WEB-MISC text/html content-type without HTML-possible malware C&C
sid:11192 POLICY download of executable content
sid:2003179ET POLICY exe download without User Agent
sid:2007671 ET POLICY Binary Download Smaller than 1 MB Likely Hostile
sid:2009033 ET POLICY Suspicious Executable (PE under 128)
sid: 2000419 ET POLICY PE EXE or DLL Windows file download

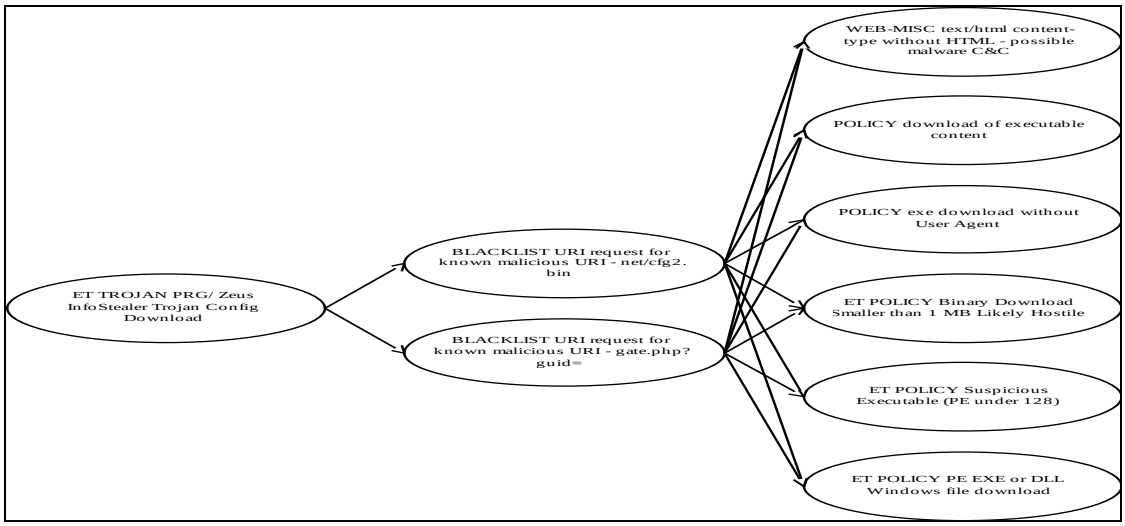

FIGURE 5. The second, third and fourth, attack stage

E. The last stage involves maintenance and update by downloading further binaries. In addition, the infected machine participates in fast scanning and visiting malicious websites that can be detected by policy rules. And in some occasions, the infected machine sends large number of DNS requests experiencing query failures or redirection which are very obvious signs of Botnet attack. This part of attack scenario is shown in Figure 6, and the whole attack graph is shown in Figure 7.

sid: 2009028 ET MALWARE 404 Response with an EXE Attached - Likely Malware Drop
sid: 2009885 ET SCAN Unusually Fast 404 Error Messages (Page Not Found), Possible Web Application Scan/Directory Guessing Attack
sid: 2011085 ET POLICY HTTP Redirect to IPv4 Address

FIGURE 6. The fifth attack stage

We have to mention that these stages can be extended to perform the main purpose of the infected machines such as DDoS, spam, and distribution of malware. These activities will be also participated in the attack map if originated from the same machine.

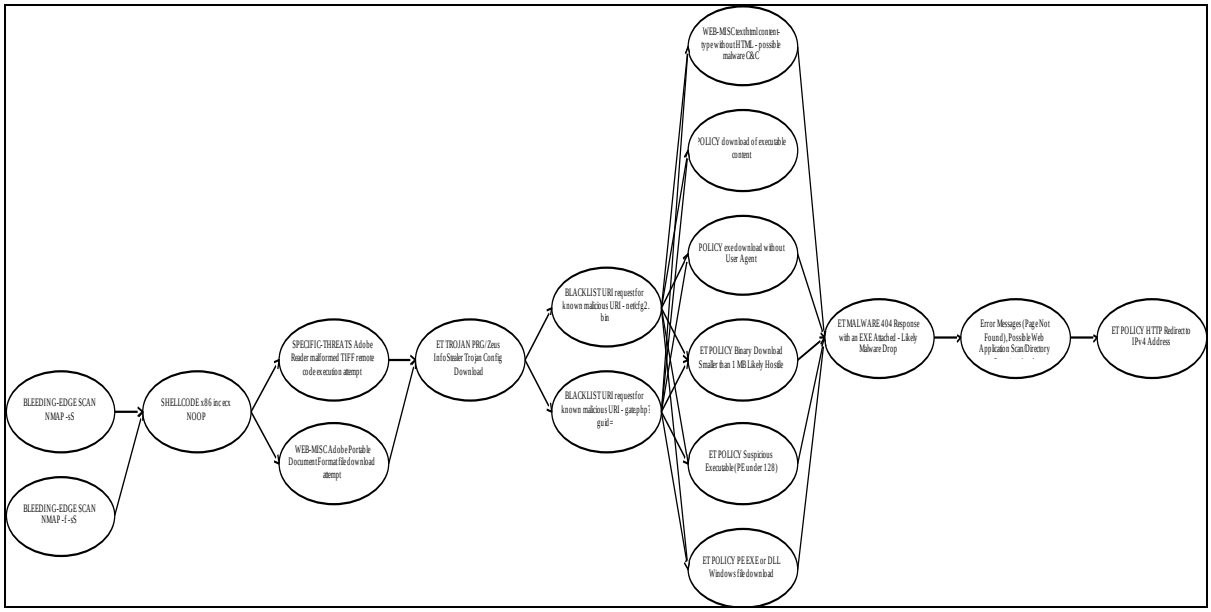

FIGURE 7. Graph of extracted Extracted Botnet scenario

VIII. CONCLUSION & FUTURE WORK

We have presented our proposed correlation model to achieve high quality recognition of multistage attack in real time. The proposed approach is mainly based on improved version of "requires/provides" model which is basically used in plan recognition models. Novel methods have been presented to overcome the limitation of current systems: vulnerability, extensional conditions, attack direction, and administrator experience. It has been demonstrated that this mechanism can applied to detect complex multi-stage attack. We have analyzed Botnet traffic as a case study to measure accuracy and performance of MARS tool which has been developed based on the proposed model. We have confidence that our system has achieved an improvement in relation to identification of attack plans and reduction in graph complexity. False positives have been reduced comparing with other approaches using vulnerability knowledge. Future work will be focusing on involvement of other NIDS and antivirus systems to broaden the volume of the supplied information. We are not concerned of a huge data as only related activities are detected ignoring false positives. In addition, implementation of statistical engine will gives more accurate results to provide some other inputs different from the signature-based systems.

REFERENCES

[1] Ai-fang Zhang, Zhi-tang Li, Dong Li, Li Wang, "Discovering Novel Multistage Attack Patterns in Alert Streams," Networking, Architecture, and Storage, In International Conference on Networking, Architecture, and Storage (NAS 2007), 2007.

[2] A. Valdes and K. Skinner. Probabilistic alert correlation. Lecture Notes in Computer Science, 2212:54-68, 2001

[3] Basic Analysis and Security Engine; http://base.secureideas.net/

[4] Baylor, K.; Brown, C. Killing Botnets: a view from the trenches. October 2006. http://www.mcafee.com/ us/local_content/white_papers/wp_botnet.pdf.

[5] B. Zhu and A. A. Ghorbani. "Alert correlation for extracting attack strategies". International Journal of Network Security, 3(2):244-258, 2006.

[6] Choi, H., Lee, H., and Kim, H. 2009. "BotGAD: detecting botnets by capturing group activities in network traffic". In Proceedings of the Fourth international ICST Conference on Communication System Software and middlewaRE (Dublin, Ireland, June 16 - 19, 2009). COMSWARE '09.

[7] Common Vulnerabilities Exposure, Aug 2010, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0188

[8] F. Alserhani, M. Akhlaq, I. Awan, A. Cullen, and P. Mirchandani, "MARS: Multi Stage Attack Recognition System", In Proc. of the International Conf. on Advanced Information Networking and Applications (AINA), Perth, Australia, 2010 , pp.753-759

[9] F. Cuppens. "Managing alerts in a multi-intrusion detection environment". In 17th Annual Computer Security Applications Conference, New-Orleans, USA, Dec 2001.

[10] F. Cuppens and R. Ortalo. Lambda: "A language to model a database for detection of attacks" . In RAID '00: Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection, pages197-216, London, UK, 2000. Springer-Verlag.

[11] F. Cuppens and A. Miege. "Alert correlation in a cooperative intrusion detection framework". In SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy, page 202, Washington, DC, USA, 2002. IEEE Computer Society.

[12] Foster, J. C. 2007, "Metasploit Toolkit for Penetration Testing, Exploit Development", and Vulnerability Research. Syngress Publishing.

[13] Jie Ma, Zhi-tang Li, Wei-ming Li, "Real-Time Alert Stream Clustering and Correlation for Discovering Attack Strategies," fskd, vol. 4, pp.379-384, 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008

[14] K. Julisch. "Clustering intrusion detection alarms to support root cause analysis". ACM Trans. Inf. Syst.Secur., 6(4):443-471, 2003.

[15] Lincoln Labs Information Systems Technology, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

[16] Li, Z., A. Zhang, et al. "Real-Time Correlation of Network Security Alerts". Proceedings of the IEEE International Conference on e-Business Engineering, IEEE Computer Society, 2007

[17] Nessus: Security Scanner; http://www.nessus.org

[18] Peng Ning, Yun Cui, Douglas Reeves, and Dingbang Xu, "Tools and Techniques for Analyzing Intrusion Alerts", in ACM Transactions on Information and System Security, 7(2): 273--318, May 2004.

[19] Peng Ning, Yun Cui, Douglas S. Reeves, "Constructing Attack Scenarios through Correlation of Intrusion Alerts," in Proceedings of the 9th ACM Conference on Computer & Communications Security, pages 245--254, Washington D.C., November 2002.

[20] R. Agrawal and R. Srikant: "Mining sequential patterns". In: Research Report RJ 9910, IBM Almaden Research Center, San Jose, California, October 1994.

[21] S. Eckmann, G. Vigna, and R. Kemmerer. "Statl: An attack language for state-based intrusion detection", Journal of Computer Security , 10(1-2):71-104 ,2002.

[22] S. J. Templeton and K. Levitt. "A requires/provides model for computer attacks". In NSPW '00: Proceedings of the 2000 workshop on New security paradigms, pages 31-38, New York, NY, USA, 2000. ACM Press.

[23] Snort: "A free lightweight network intrusion detection system for UNIX and Windows"; http://www.snort.org/

[24] Stinson, E. and Mitchell, J. C. 2008. "Towards systematic evaluation of the evadability of bot/botnet detection methods". In Proceedings of the 2nd Conference on USENIX Workshop on offensive Technologies (San Jose, CA). USENIX Association, Berkeley, CA, 1-9

[25] Wang, L., A. Liu, et al. (2005). "An efficient and unified approach to correlating, hypothesizing, and predicting intrusion alerts". Computer Security–ESORICS 2005, No. 3679, pages 247-266

[26] X. Qin. A Probabilistic-Based Framework for INFOSEC Alert Correlation. PhD thesis, Georgia Institute of Technology, 2005.

[27] X. Qin and W. Lee. "Attack plan recognition and prediction using causal networks". In ACSAC '04: Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04), pages 370-379, Washington, DC, USA, 2004. IEEE Computer Society.

[28] Zeidanloo, H., A. Manaf, et al. "A Proposed Framework for P2P Botnet Detection". In IACSIT International Journal of Engineering and Technology, Vol.2, No.2, April 2010

[29] Zeus Tracker, Aug 2010, https://zeustracker.abuse.ch

[30] Zhi-tang Li, Jie Lei, Li Wang, Dong Li, "A Data Mining Approach to Generating Network Attack Graph for Intrusion Prediction," fskd, vol. 4, pp.307-311, Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007) Vol.4, 2007